

LẬP TRÌNH MẠNG CĂN BẢN

Biên soạn: ThS. Đỗ Thị Hương Lan



TRƯỜNG ĐH CÔNG NGHỆ THÔNG TIN - ĐHQG-HCM
KHOA MẠNG MÁY TÍNH & TRUYỀN THÔNG
FACULTY OF COMPUTER NETWORK AND COMMUNICATIONS

Tầng 8 - Tòa nhà E, trường ĐH Công nghệ Thông tin, ĐHQG-HCM

Chương 8

Lập trình với Giao thức truyền tải file FTP

Nội dung chi tiết

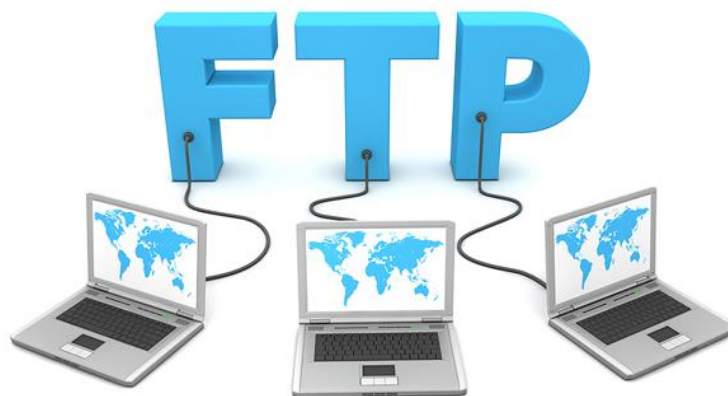
- Giới thiệu về FTP và nguyên tắc hoạt động
- Cài đặt FTP Server và FTP Client
- Lệnh FTP trong CMD của Windows
- Lập trình FTP Client

Nội dung chi tiết

- Giới thiệu về FTP và nguyên tắc hoạt động
- Cài đặt FTP Server và FTP Client
- Lệnh FTP trong CMD của Windows
- Lập trình FTP Client

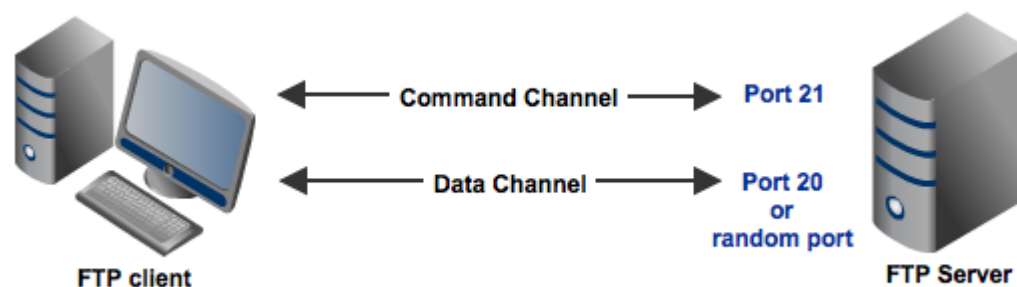
Giới thiệu

- Để website chạy cần phải đưa source web đã thiết kế lên server
- Quá trình đưa lên thông qua giao thức truyền file (Ví dụ: FTP)
- FTP là cơ chế chính để truyền file giữa các máy tính thông qua môi trường Internet

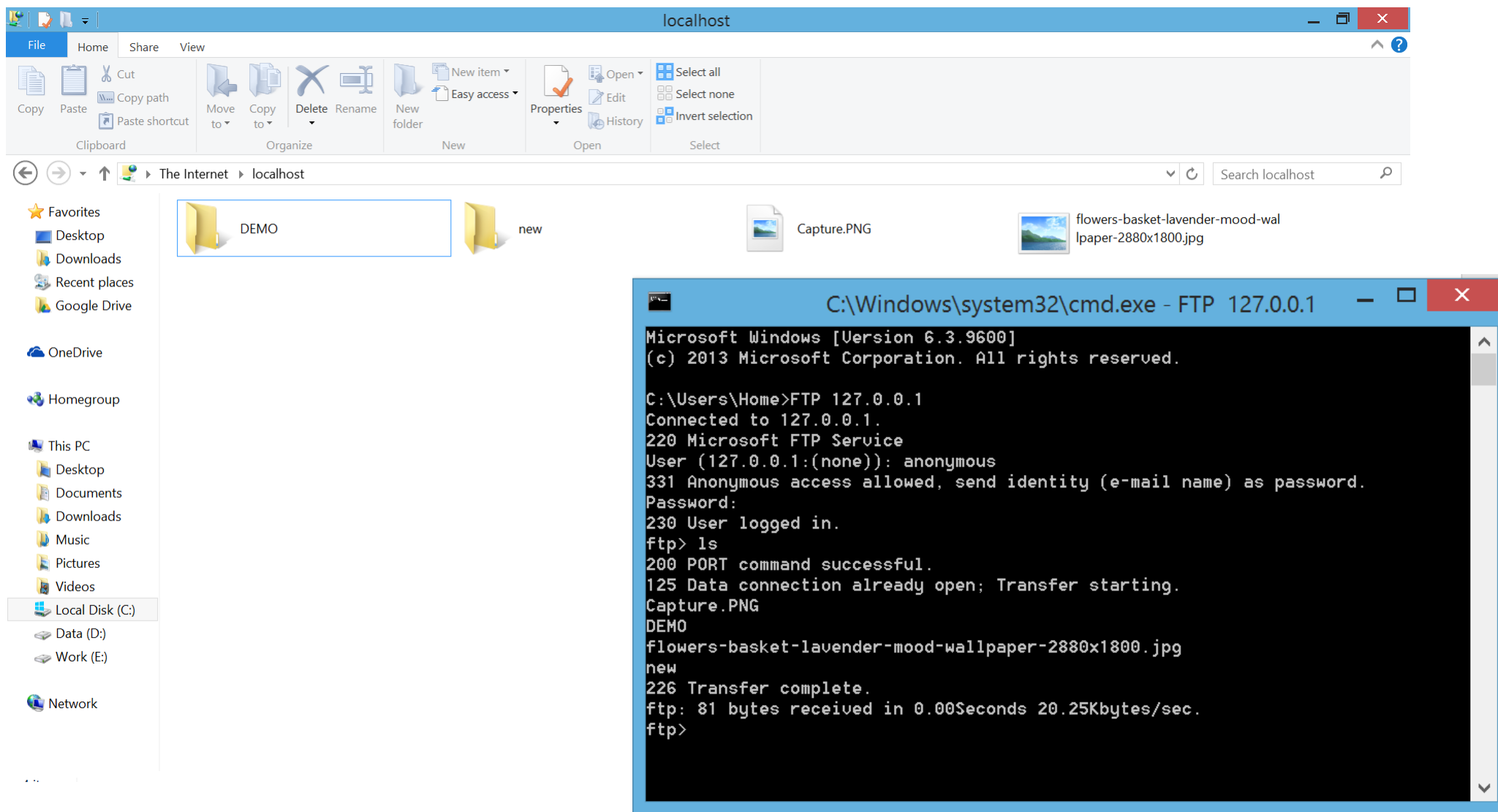


Tổng quan về FTP

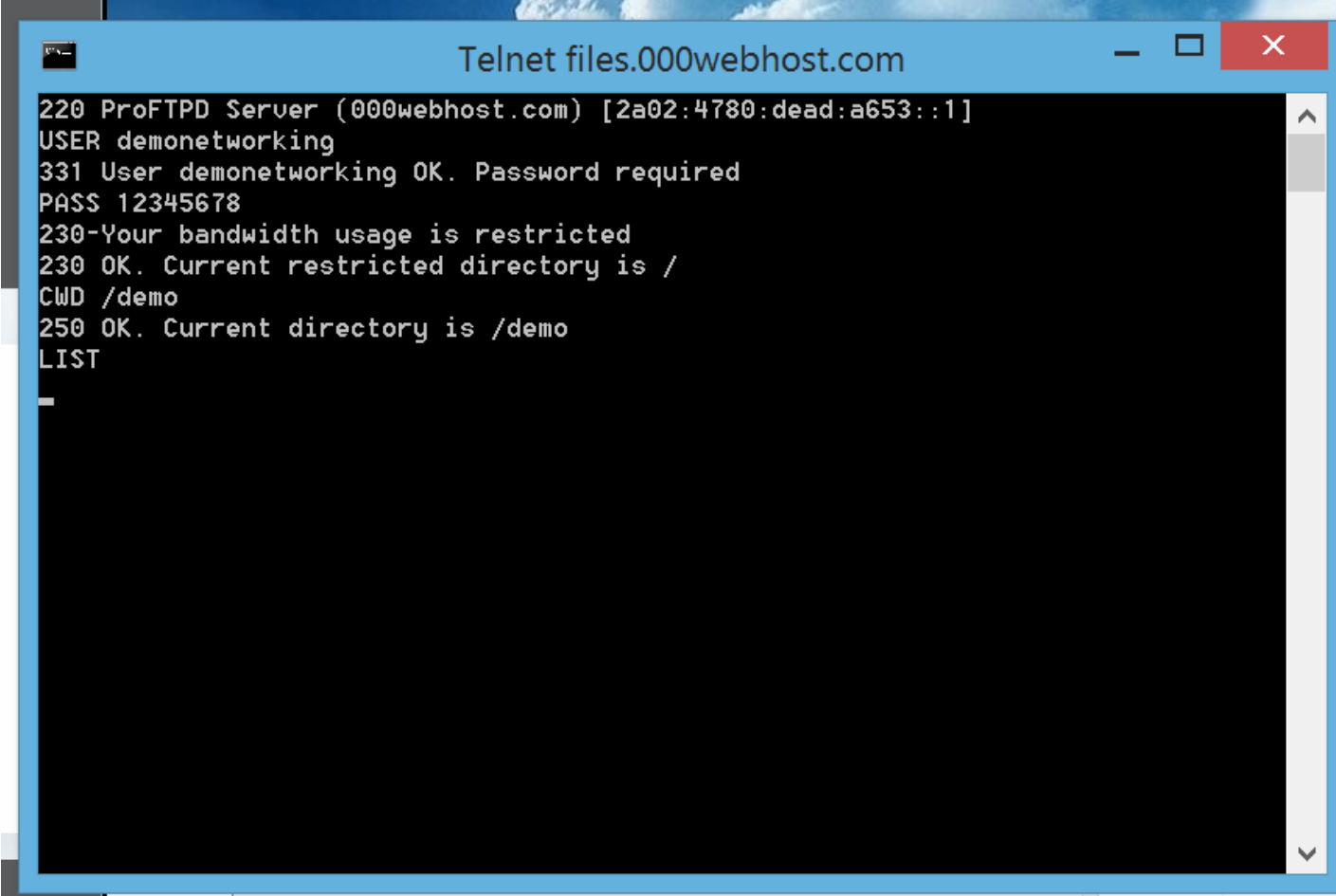
- FTP hoạt động trên 2 port:
 - 21 cho socket điều khiển
 - 20 cho socket dữ liệu
- FTP được định nghĩa trong RFC 959
- Lệnh giao dịch giữa client và server tương đối dễ hiểu và ngắn thành các dòng



Ví dụ



FTP COMMANDS WITH TELNET



```
Telnet files.000webhost.com
220 ProFTPD Server (000webhost.com) [2a02:4780:dead:a653::1]
USER demonetworking
331 User demonetworking OK. Password required
PASS 12345678
230-Your bandwidth usage is restricted
230 OK. Current restricted directory is /
CWD /demo
250 OK. Current directory is /demo
LIST
```


Mã trạng thái FTP

Mã	Ý nghĩa
1xx	Phản hồi bước đầu thành công. Lệnh đã bắt đầu thực thi trên server
2xx	Phản hồi bước đầu thành công. Lệnh đã hoàn tất trên server
3xx	Phản hồi trung gian thành công. Lệnh đã chấp nhận, nhưng chưa thực thi trên server
4xx	Phản hồi không thành công tạm thời. Lệnh đã bị từ chối, nhưng có thể được kích hoạt lại trên server
5xx	Phản hồi không thành công hoàn toàn. Lệnh đã bị từ chối, và không thể được kích hoạt lại trên server

Mã trạng thái FTP

- Ví dụ:
220 Serv-U FTP-Server v2.5k for WinSock ready...
USER secret
331 User name okay, need password.
PASS (hidden)
230 User logged in, proceed.
PWD
257 "/" is current directory.
TYPE A
200 Type set to A.
PASV
227 Entering Passive Mode (212,17,38,3,11,144)
LIST -aL
150 Opening ASCII mode data connection for /bin/ls.
226 Transfer complete.

Cách dùng các port

- Khi soạn email, kết thúc nội dung của email được đánh dấu bằng chuỗi ký tự <enter>.<enter>. Chuỗi này sẽ được loại bỏ khi gửi đi
- Với FTP, những chuỗi ký tự này có thể nằm trong nội dung, nếu gỡ bỏ có thể làm hỏng file. Để tránh vấn đề trên, FTP dùng port 21 để gửi/nhận lệnh, mỗi lệnh kết thúc bởi <enter>

Cách dùng các port

- Dữ liệu được gửi qua lại giữa client và server trên một kết nối tạm thời ở port 20, được mở khi có dữ liệu và đóng khi kết thúc truyền dữ liệu đó
- Hầu hết FTP client đóng vai trò yêu cầu và FTP server đóng vai trò phục vụ
- FTP ở chế độ Passive: client hướng dẫn server lắng nghe trên port dữ liệu không phải port mặc định là 20

Cách dùng các port

- Phản hồi cho lệnh PASV sẽ luôn luôn chứa danh sách 6 số đặt trong dấu ngoặc được phân tách bởi dấu phẩy:
 - 4 số đầu đại diện cho địa chỉ IP
 - 2 số cuối thể hiện port để server lắng nghe dữ liệu
- Ví dụ: 192,168,5,7,4,91
 - Server có địa chỉ IP là 192.168.5.7
 - Đang lắng nghe trên port 1115 ($= 4 \times 256 + 91$)

Cách dùng các port

- Server sẽ thực hiện lắng nghe trên port sớm nhất có thể sau khi nhận lệnh PASV
- Sever trả lời 227 cho biết đang lắng nghe trên port đó
- Khi client kết nối vào port này, server trả lời bằng thông điệp 150
- Nếu client không kết nối trong khoảng thời gian quy định (khoảng vài giây) → server sẽ gửi thông điệp timeout 425

Cách dùng các port

- Server sẽ gửi dữ liệu yêu cầu trên port đã mở và đóng kết nối lại sau khi hoàn thành gửi dữ liệu xong, phát ra thông điệp 226
- Tiến trình tương tự diễn ra theo thứ tự ngược khi upload lên server. Khi lệnh PASV được phát ra và client kết nối vào port do server chỉ định. Client sau đó đặt nội dung file lên socket mới và đóng lại sau khi gửi file xong

FTP: tiến trình bắt tay

- Cơ chế chứng thực:
FTP chấp nhận username / password dạng text thô → có thể nhìn thấy được với chương trình phân tích
- FTP trên SSL được khuyến cáo khi website truyền thông tin quan trọng

FTP: tiến trình bắt tay

- FTP server cũng cho phép truy cập ẩn danh (anonymous). Khi ấy username là anonymous và password tùy ý. Đây là thiết lập mặc định của dịch vụ Microsoft FTP
- Khi kết nối vào FTP server ở port 21, server sẽ phản hồi như sau:
220 <some message><enter>

FTP: tiến trình bắt tay

- Giống quá trình bắt tay của POP3, các lệnh USER và PASS được gửi lần lượt:
USER <username><enter>
- Server phản hồi:
331 <some message><enter>
- Client gửi:
PASS <password><enter>

FTP: tiến trình bắt tay

- Server phản hồi nếu đăng nhập thành công:
230 <some message><enter>
- Server phản hồi nếu đăng nhập thất bại:
530 <some message><enter>
- Nếu đăng nhập thành công thì người dùng có thể truy cập vào FTP server với quyền tương ứng

FTP: tiến trình bắt tay

- Một số FTP server sẽ hủy kết nối với người dùng không có thao tác gì để tiết kiệm tài nguyên
- Dùng lệnh NOOP để báo cho server biết nhằm tránh hiện tượng trên

NOOP<enter>

200 <message><enter>

FTP: tiến trình bắt tay

- Để đóng kết nối, client đơn giản chỉ cần đóng kết nối TCP bên dưới bằng cách gửi lệnh QUIT
QUIT<enter>
221 <message><enter>

FTP COMMANDS

5.3.1. FTP COMMANDS

The following are the FTP commands:

```
USER <SP> <username> <CRLF>
PASS <SP> <password> <CRLF>
ACCT <SP> <account-information> <CRLF>
CWD <SP> <pathname> <CRLF>
CDUP <CRLF>
SMNT <SP> <pathname> <CRLF>
QUIT <CRLF>
REIN <CRLF>
PORT <SP> <host-port> <CRLF>
PASV <CRLF>
TYPE <SP> <type-code> <CRLF>
STRU <SP> <structure-code> <CRLF>
MODE <SP> <mode-code> <CRLF>
RETR <SP> <pathname> <CRLF>
STOR <SP> <pathname> <CRLF>
STOU <CRLF>
APPE <SP> <pathname> <CRLF>
ALLO <SP> <decimal-integer>
    [<SP> R <SP> <decimal-integer>] <CRLF>
REST <SP> <marker> <CRLF>
RNFR <SP> <pathname> <CRLF>
RNTD <SP> <pathname> <CRLF>
ABOR <CRLF>
DELE <SP> <pathname> <CRLF>
RMD <SP> <pathname> <CRLF>
MKD <SP> <pathname> <CRLF>
PWD <CRLF>
LIST [<SP> <pathname>] <CRLF>
NLST [<SP> <pathname>] <CRLF>
SITE <SP> <string> <CRLF>
SYST <CRLF>
STAT [<SP> <pathname>] <CRLF>
HELP [<SP> <string>] <CRLF>
NOOP <CRLF>
```

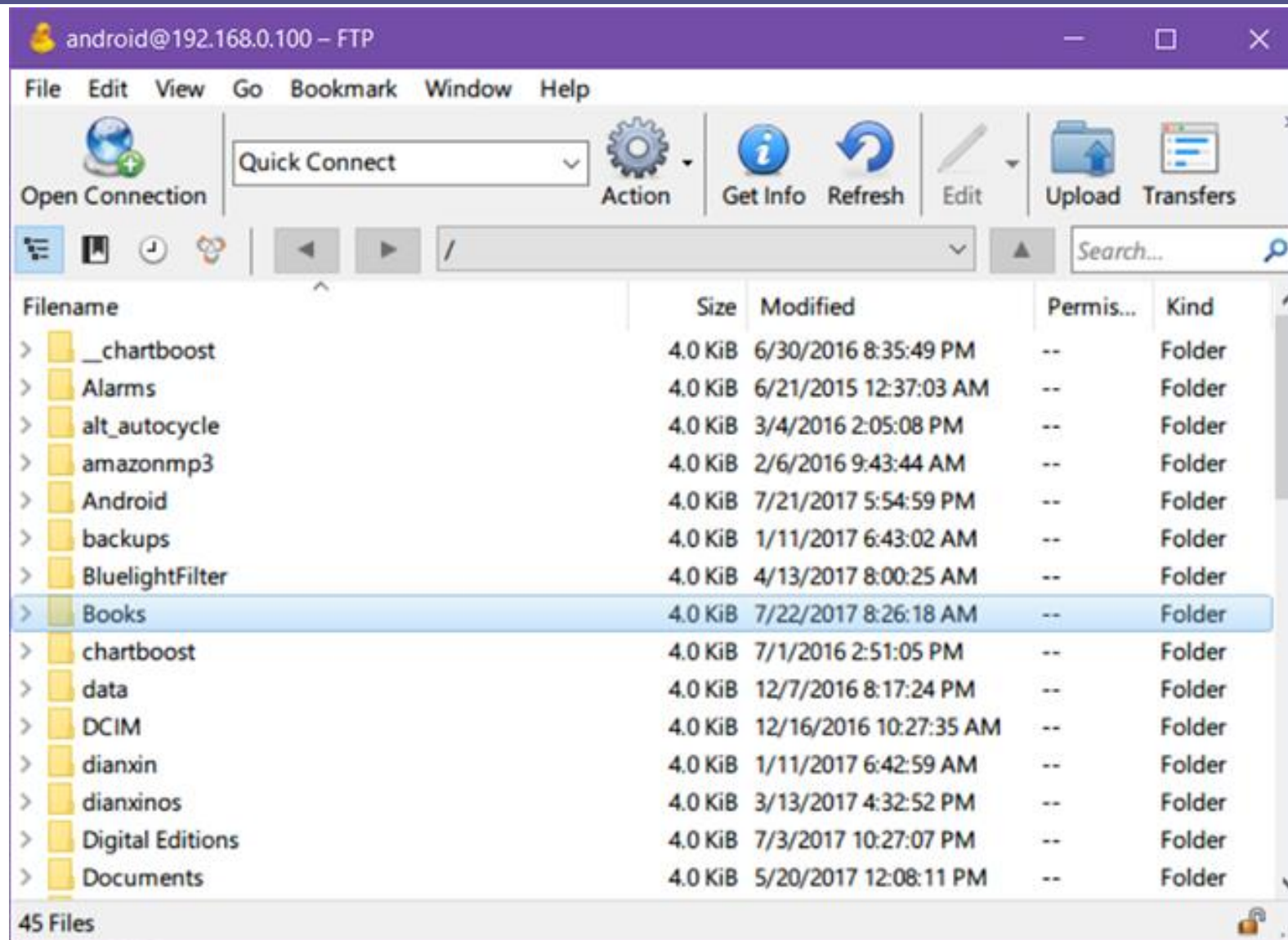
Nội dung chi tiết

- Giới thiệu về FTP và nguyên tắc hoạt động
- Cài đặt FTP Server và FTP Client
- Lệnh FTP trong CMD của Windows
- Lập trình FTP Client

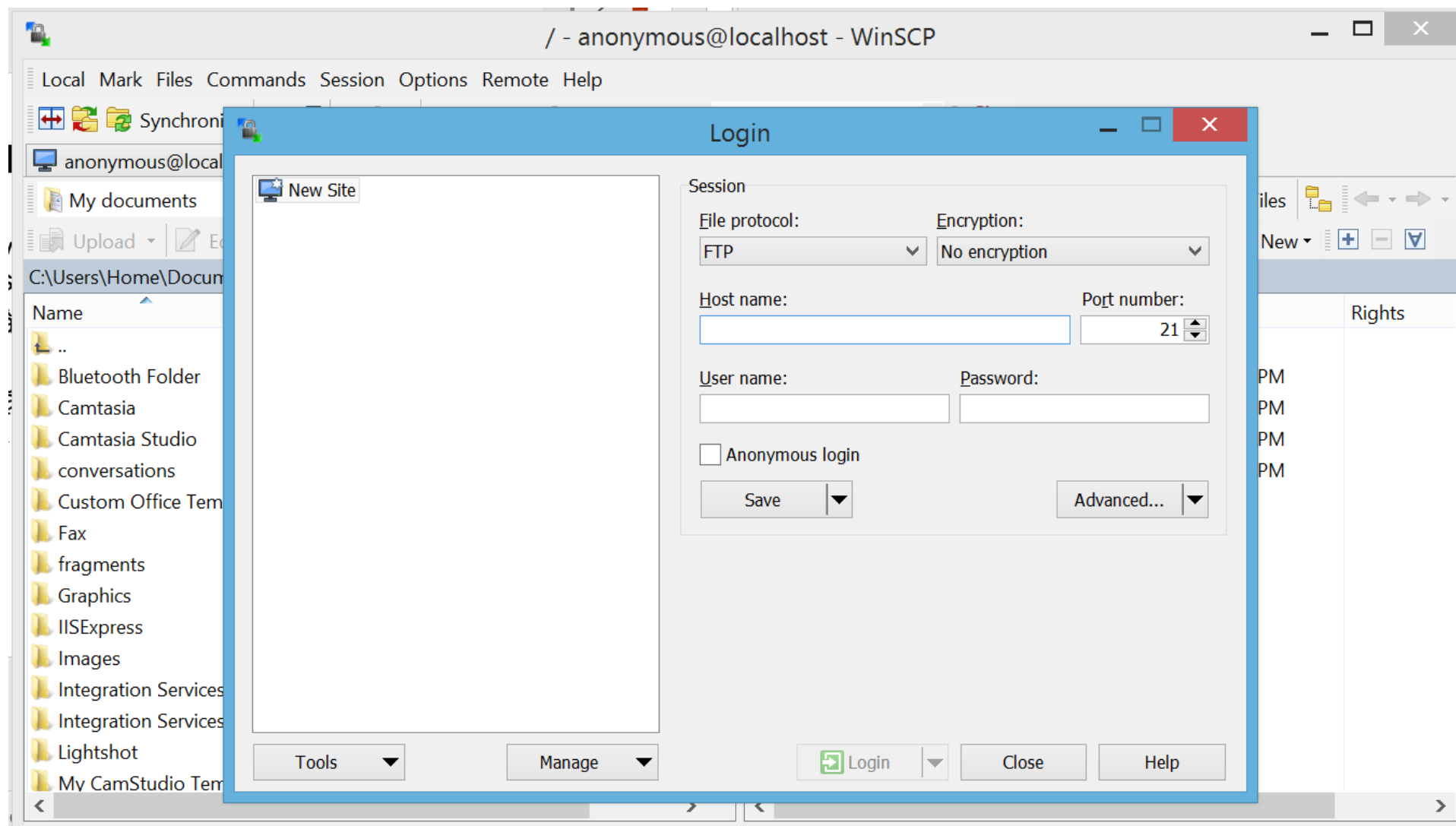
Cài đặt FTP server

- Để truy cập vào FTP server, cần biết địa chỉ IP và phải có username/password
- Trước tiên cần thực hiện cài đặt FTP server
- Có thể sử dụng công cụ hỗ trợ sẵn trong Windows hoặc cài đặt các phần mềm khác như: FileZilla, WinSCP

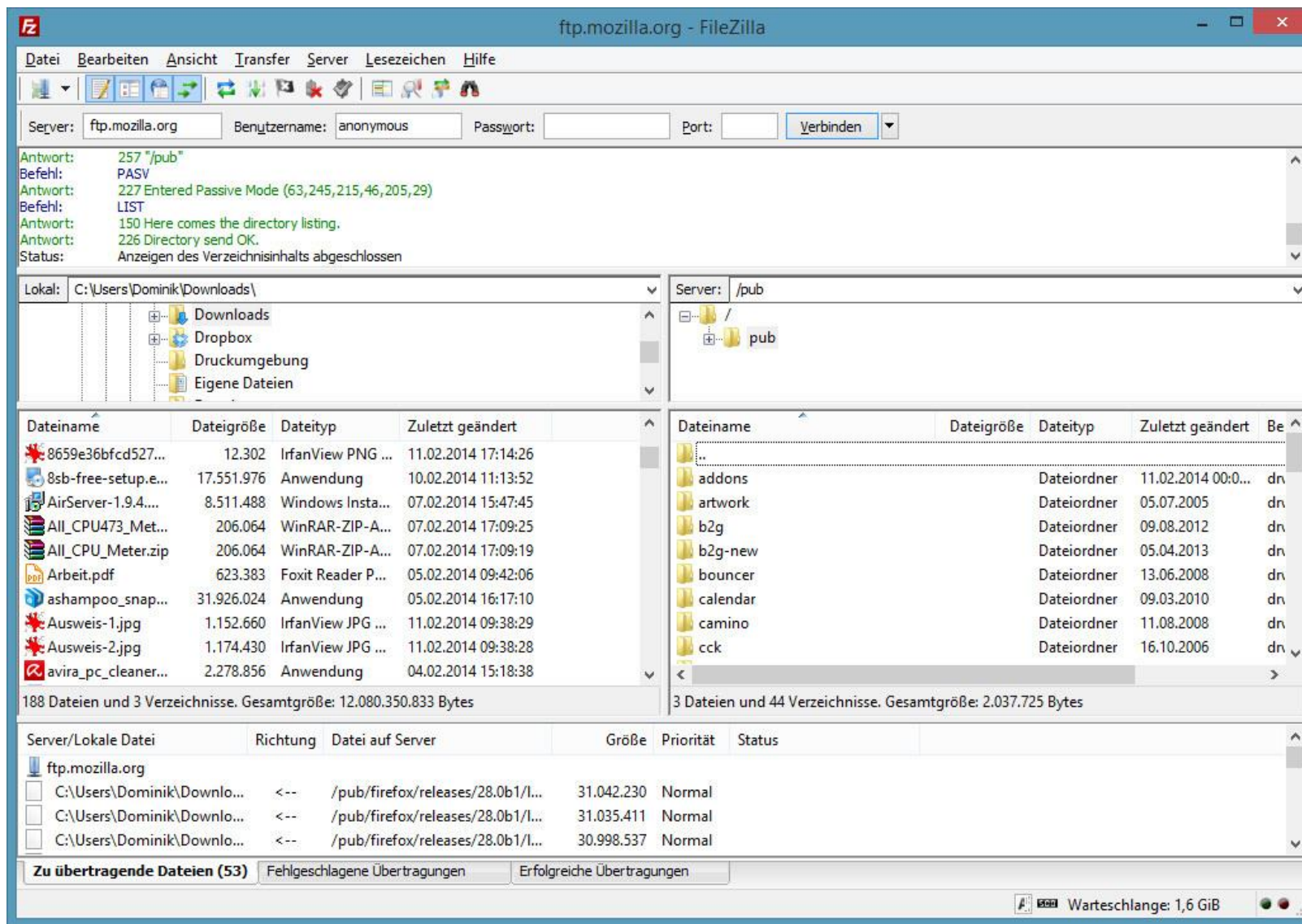
FTP Client – Ví dụ



FTP Client - WinSCP



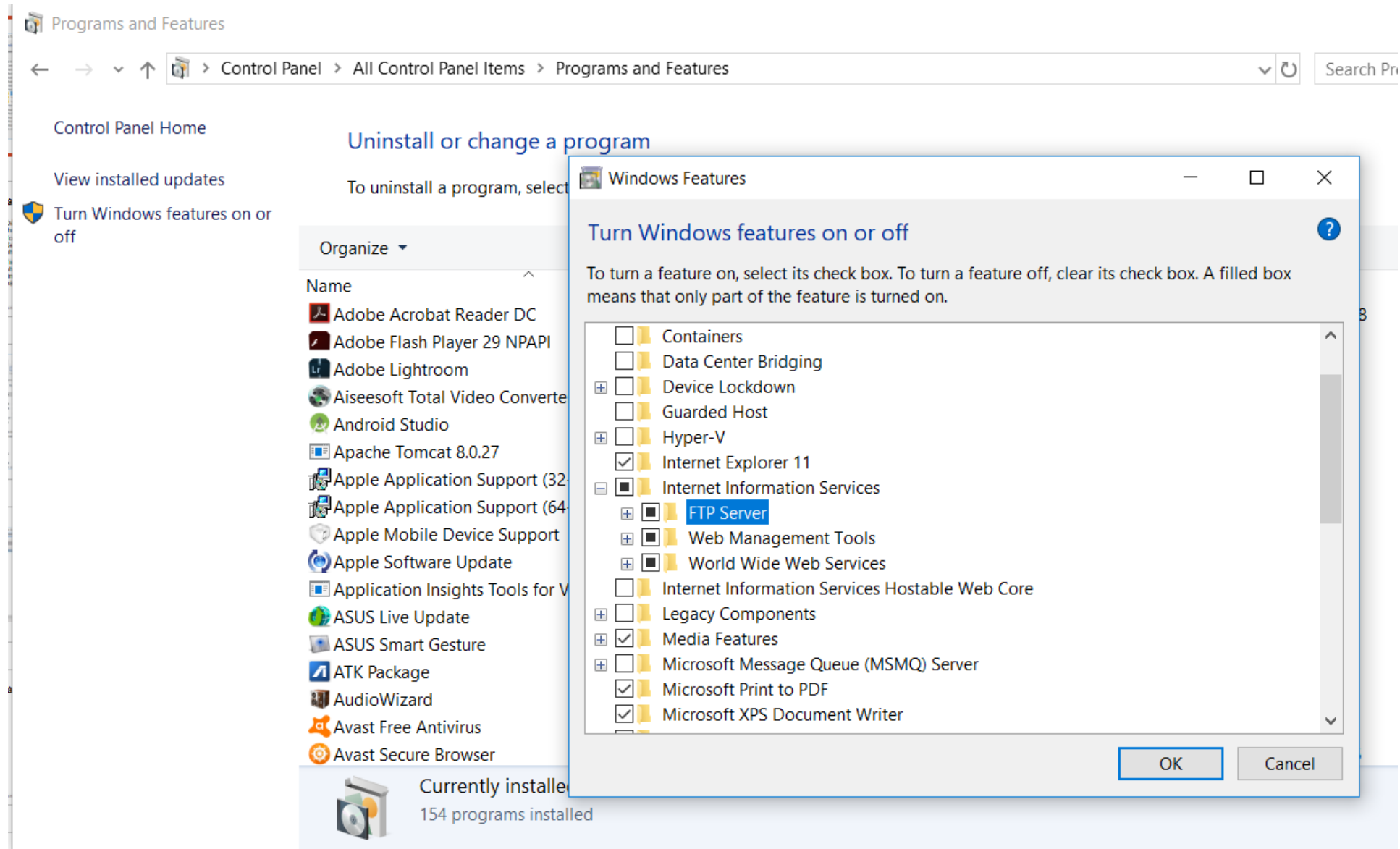
FTP Client - FileZilla



Cài đặt FTP server trên Windows

Control Panel → Add/Remove Programs → Add or Remove Windows Components → Internet Information Services → FTP Service

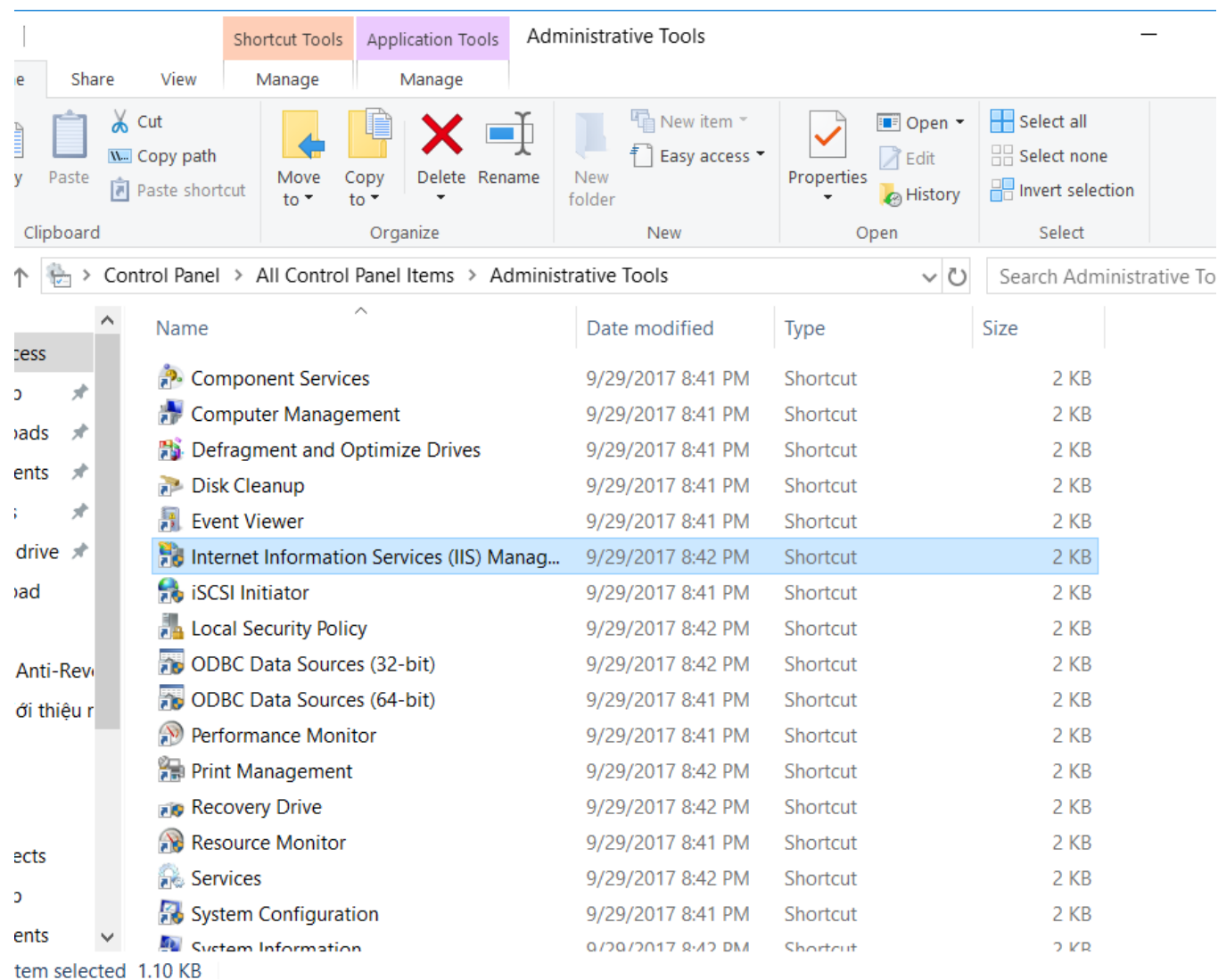
Cài đặt FTP server trên Windows



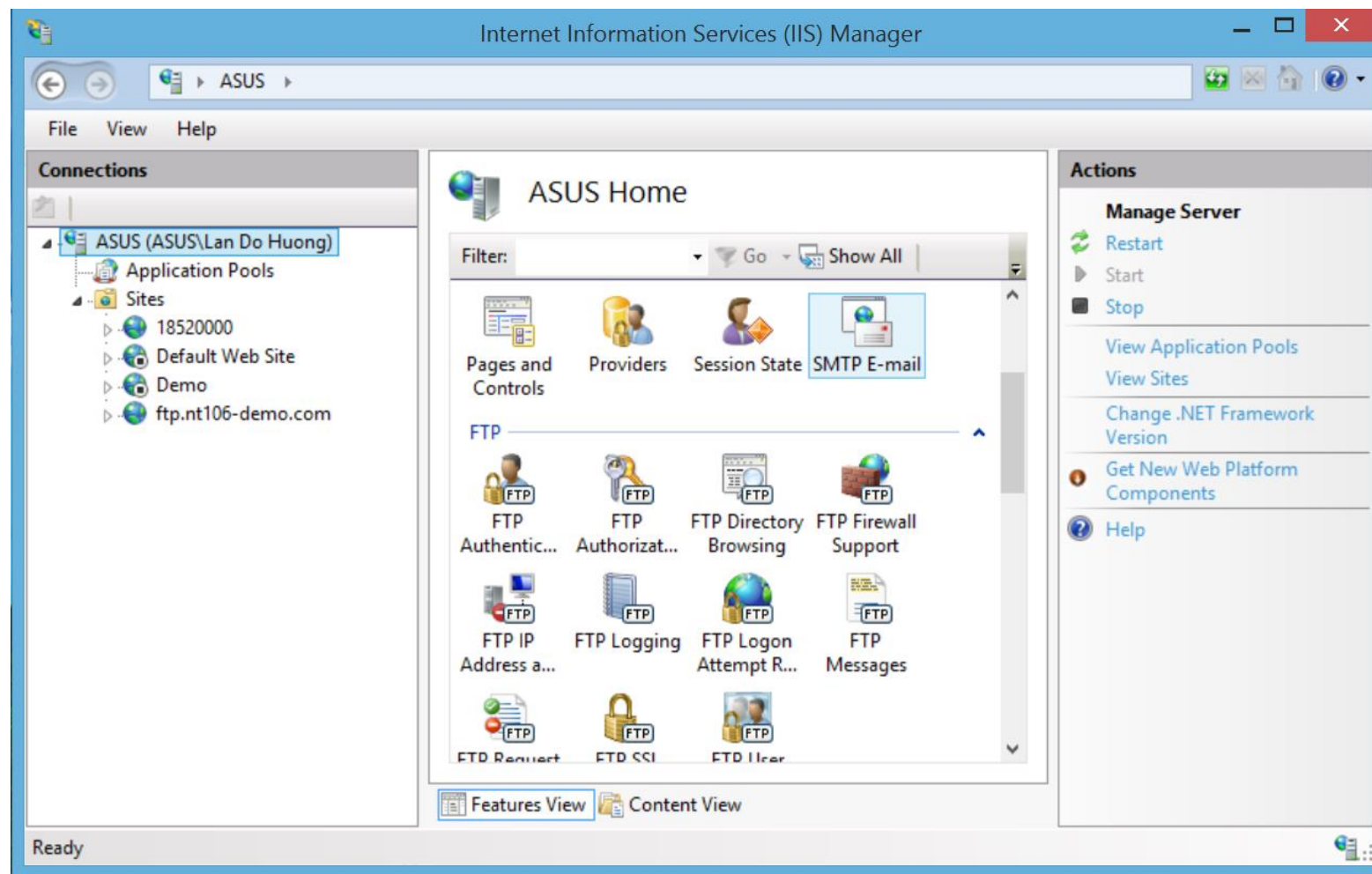
Cài đặt FTP trên Windows

- Để quản trị và tinh chỉnh cấu hình FTP server: Control Panel → Administrative Tools → Internet Information Services → FTP
- Hoặc Internet Information Services → Add FTP Site
- Chỉnh sửa các thiết lập:
 - Cho phép số kết nối đồng thời tối đa
 - Thời gian timeout
 - Thư mục vật lý
 - Cho phép đăng nhập nặc danh,...

Cài đặt FTP server trên Windows



Cài đặt FTP server trên Windows



Cài đặt FTP trên Windows

The screenshot shows the Internet Information Services (IIS) Manager console. The left pane displays the tree view with the path: ASUS > Sites > ftp.nt106-demo.com. The main pane shows the 'ftp.nt106-demo.com Home' directory with a filter set to 'FTP'. A red circle highlights the 'FTP Authentication' icon. An inset window titled 'FTP Authentication' is open, showing a table with authentication modes and their status.

Mode	Status	Type
Anonymous Authentication	Enabled	Built-In
Basic Authentication	Enabled	Built-In

The status 'Enabled' for both authentication modes is circled in red in the original image.

Nội dung chi tiết

- Giới thiệu về FTP và nguyên tắc hoạt động
- Cài đặt FTP Server và FTP Client
- **Lệnh FTP trong CMD của Windows**
- Lập trình FTP Client

Lệnh FTP trong WINDOWS CMD

Lệnh	Ý nghĩa
?	Giúp đỡ
ascii	chuyển chế độ truyền là ASCII
binary	chuyển chế độ truyền là nhị phân
bye	Thoát khỏi FTP (giống lệnh quit)
cd	Thay đổi thư mục trên máy chủ
close	Kết thúc kết nối đến máy tính khác
close brubeck	Đóng kết nối hiện tại với máy tính khác, nhưng vẫn giữ môi trường lệnh FTP
delete	xóa 1 file trong thư mục từ xa hiện hành (trên server)
get	Để copy 1 file từ máy từ xa về máy cục bộ
get ABC DEF	Copy file ABC trong thư mục từ xa hiện hành lên file DEF trong máy cục bộ (ghi đè).

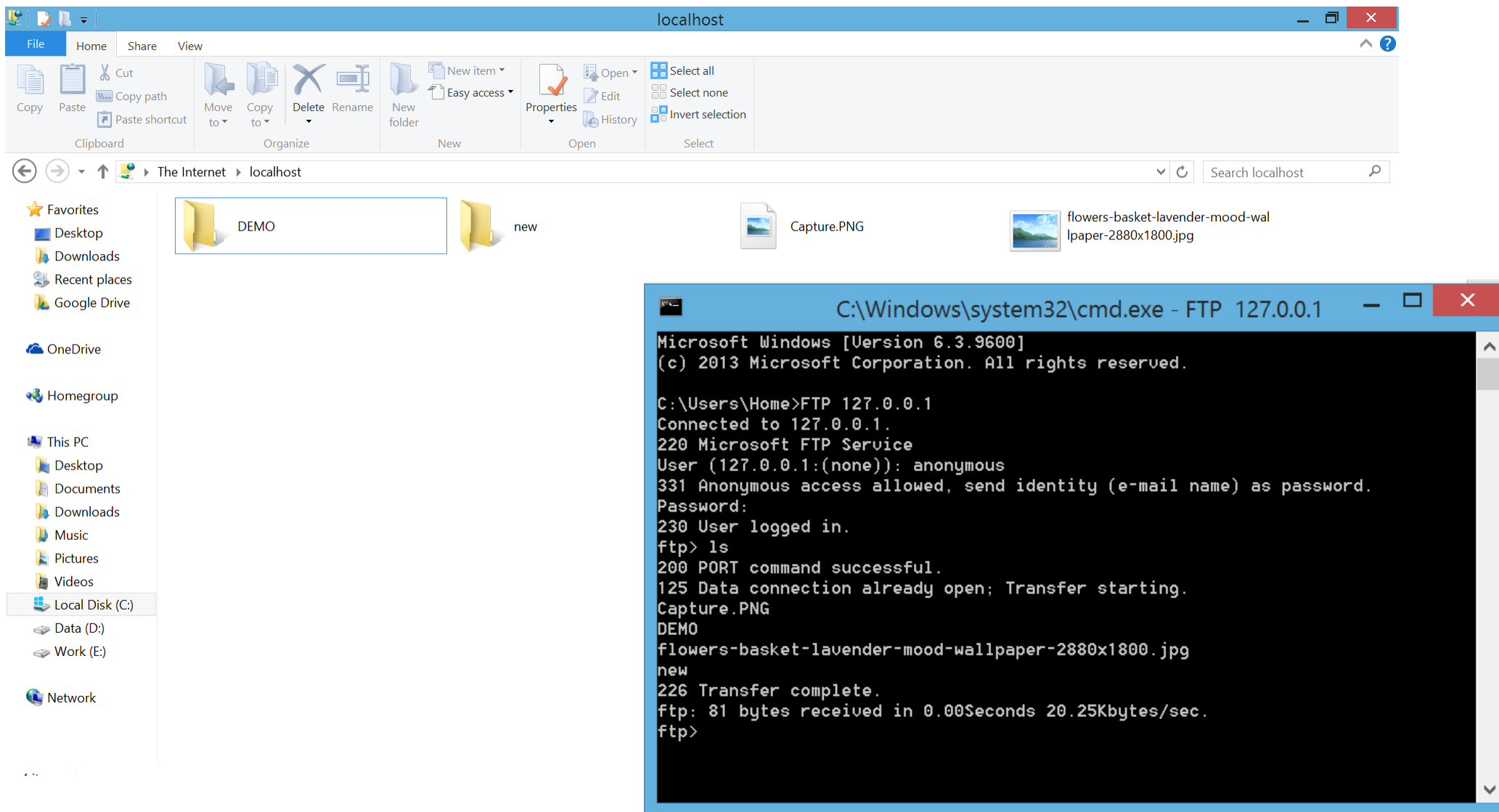
Lệnh FTP trong WINDOWS CMD

Lệnh	Ý nghĩa
get ABC	Copy file ABC trong thư mục từ xa hiện hành lên file ABC trong máy cục bộ (cùng tên).
help	Xem danh sách các lệnh FTP sẵn sàng
lcd	thay đổi thư mục trên máy ở xa
ls	Liệt kê tên các file trên thư mục hiện hành trong máy ở xa
mkdir	Tạo thư mục mới trong máy ở xa
mget	Copy nhiều file từ máy ở xa về máy hiện hành; sẽ được nhắc y/n cho mỗi file
mget *	Copy tất cả các file từ máy ở xa về máy hiện hành; sẽ được nhắc y/n cho mỗi file. Dùng ký tự đại diện *.
mput	Copy nhiều file từ máy hiện hành lên máy ở xa; sẽ được nhắc y/n cho mỗi file
open	Mở 1 kết nối với máy tính khác

Lệnh FTP trong WINDOWS CMD

Lệnh	Ý nghĩa
open brubeck	Mở 1 kết nối mới với máy tính, dùng chế độ brubeck;
put	Copy 1 file từ máy hiện hành lên máy ở xa
pwd	Xem thư mục hiện hành trên máy ở xa
quit	Thoát khỏi môi trường FTP (giống lệnh bye)
rmdir	Xóa thư mục hiện hành trên máy ở xa

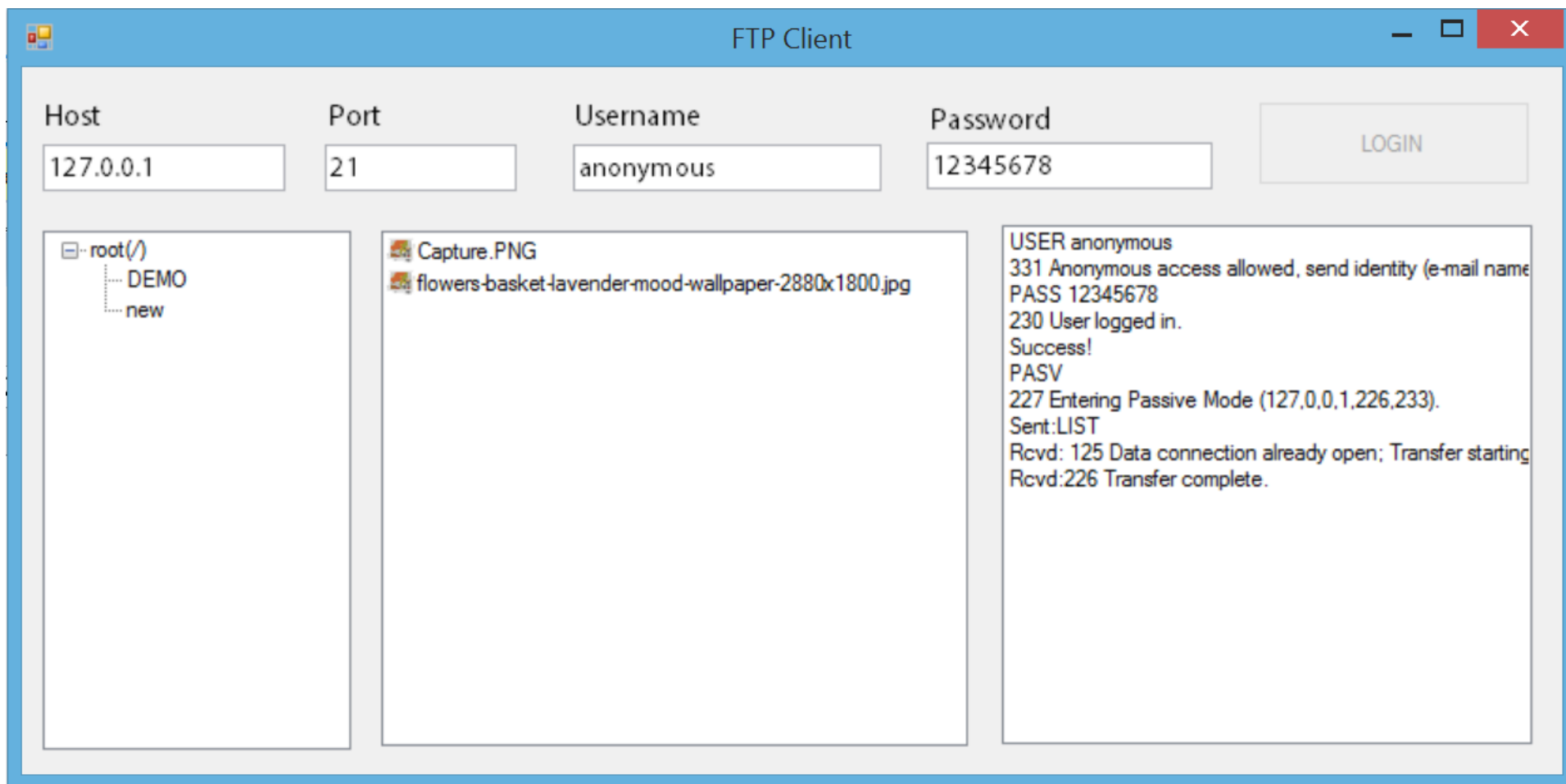
Lệnh FTP trong WINDOWS CMD: Ví dụ



Nội dung chi tiết

- Giới thiệu về FTP và nguyên tắc hoạt động
- Cài đặt FTP Server và FTP Client
- Lệnh FTP trong CMD của Windows
- Lập trình FTP Client

Lập trình FTP sử dụng tcpclient



Lập trình FTP sử dụng tcpclient

- Khai báo các biến public trong frmMain:
public const string STR_Constant = "\r\n";
public NetworkStream NetStrm;
public string RemotePath = "";
public string server = "";
- Đối tượng của lớp NetworkStream dùng để kết nối lệnh FTP (port 21)
- Kết nối dữ liệu không bền vững (truyền xong đóng lại ngay) nên không cần khai báo toàn cục

Lập trình FTP sử dụng tcpclient

```
TcpClient tcpClient = new TcpClient();
tcpClient.Connect(ipAddress, port);
StreamWriter sw = new StreamWriter(tcpClient.GetStream());
NetStrm = tcpClient.GetStream();
StreamReader sr = new StreamReader(NetStrm);
string result = sr.ReadLine();
// MessageBox.Show(result);
data = sendFTPcmd("USER " + txtUsername.Text.Trim() + STR_Constant);
data = sendFTPcmd("PASS " + txtPass.Text + STR_Constant);
if (data.Substring(0, 3) != "230")
{
    ListViewItem lv = new ListViewItem("Failed to log in!");
    lvServer.Items.Add(lv);
}
else
{
    ListViewItem lv = new ListViewItem("Success!");
    lvServer.Items.Add(lv);
}
```

Lập trình FTP sử dụng TCPClient

- Phương pháp cơ bản là tạo một đối tượng NetStrm thuộc lớp NetworkStream để mở ra socket. Các lệnh FTP (dưới dạng string) được gửi qua socket này
- Một đối tượng của lớp StreamReader được tạo ra nhằm đọc nội dung gửi trên stream này.

Lập trình FTP sử dụng tcpclient

```
public string sendFTPcmd(string cmd)
{
    byte[] szData;
    string returnedData = "";
    StreamReader RdStrm = new StreamReader(NetStrm);
    szData = Encoding.ASCII.GetBytes(cmd.ToCharArray());
    NetStrm.Write(szData, 0, szData.Length);
    returnedData = RdStrm.ReadLine();
    ListViewItem lv1 = new ListViewItem(cmd);
    lvServer.Items.Add(lv1);
    ListViewItem lv = new ListViewItem(returnedData);
    lvServer.Items.Add(lv);
    return returnedData;
}
```

Lập trình FTP sử dụng tcpclient

```
private NetworkStream createPassiveConnection()
{
    string[] commaSeperatedValues;
    server = txtHost.Text.Trim();
    int highByte = 0;
    int lowByte = 0;
    int passivePort = 0;
    string response = "";
    TcpClient clientSocket;
    NetworkStream pasvStrm = null;
    response = sendFTPCmd("PASV\r\n");
    // 227 Entering Passive Mode (127,0,0,1,4,147).
    commaSeperatedValues = response.Split(",".ToCharArray());
    highByte = Convert.ToInt16(commaSeperatedValues[4]) * 256;
    commaSeperatedValues[5] =
    commaSeperatedValues[5].Substring(0,
    commaSeperatedValues[5].IndexOf(")"));
    lowByte = Convert.ToInt16(commaSeperatedValues[5]);
    passivePort = lowByte + highByte;
    clientSocket = new TcpClient(server, passivePort); //TcpClient
    pasvStrm = clientSocket.GetStream(); //NetworkStream
    return pasvStrm;
}
```

Lập trình FTP sử dụng tcpclient

```
public byte[] sendPassiveFTPCmd(string cmd)
{
    byte[] szData;
    System.Collections.ArrayList al = new ArrayList();
    byte[] RecvBytes = new byte[Byte.MaxValue];
    Int32 bytes;
    Int32 totalLength = 0;
    szData = System.Text.Encoding.ASCII.GetBytes(cmd.ToCharArray());
    NetworkStream passiveConnection;
    passiveConnection = createPassiveConnection();
    ListViewItem lv1 = new ListViewItem("\r\nSent:" + cmd);
    lvServer.Items.Add(lv1);
    StreamReader commandStream = new StreamReader(NetStrm);
    NetStrm.Write(szData, 0, szData.Length);
    while (true)
    {
        bytes = passiveConnection.Read(RecvBytes,
            0, RecvBytes.Length);
        if (bytes <= 0) break;
        totalLength += bytes;
        al.AddRange(RecvBytes);
    }
    al = al.GetRange(0, totalLength);
    ListViewItem lv2 = new ListViewItem("\r\nRcvd: " + commandStream.ReadLine());
    lvServer.Items.Add(lv2);
    ListViewItem lv3 = new ListViewItem("\r\nRcvd:" + commandStream.ReadLine());
    lvServer.Items.Add(lv3);
    return (byte[])al.ToArray((new byte()).GetType());
}
```

Hỗ trợ FTP trên .NET 2.0 trở lên

- .NET 2.0 trở lên hỗ trợ FTP thông qua class WebClient, FtpWebRequest
- Cho phép lập trình ở mức socket hoặc thông qua các đối tượng COM

Hỗ trợ FTP trong WebClient

```
public void downloadFile()
{
    string filename = "ftp://ms.com/files/dotnetfx.exe";
    WebClient client = new WebClient();
    client.DownloadFile(filename, "dotnetfx.exe");
}
```


Hỗ trợ FTP trong FtpWebRequest

<https://docs.microsoft.com/en-us/dotnet/framework/network-programming/ftp>

<https://docs.microsoft.com/en-us/dotnet/framework/network-programming/how-to-download-files-with-ftp>

<https://docs.microsoft.com/en-us/dotnet/framework/network-programming/how-to-upload-files-with-ftp>

<https://docs.microsoft.com/en-us/dotnet/framework/network-programming/how-to-list-directory-contents-with-ftp>

Hỗ trợ FTP trong FtpWebRequest

```
private void Upload(string filename)
{
    FileInfo fileInf = new FileInfo(filename);
    string uri = "ftp://" + ftpServerIP + "/" + fileInf.Name;
    FtpWebRequest reqFTP;
    reqFTP = (FtpWebRequest)FtpWebRequest.Create(new Uri("ftp://" + ftpServerIP + "/" + fileInf.Name));
    reqFTP.Credentials = new NetworkCredential(ftpUserID, ftpPassword);
    reqFTP.KeepAlive = false;
    reqFTP.Method = WebRequestMethods.Ftp.UploadFile;
    reqFTP.UseBinary = true;
    reqFTP.ContentLength = fileInf.Length;
    int buffLength = 2048;
    byte[] buff = new byte[buffLength];
```

Hỗ trợ FTP trong FtpWebRequest

```
int contentLen;
FileStream fs = fileInf.OpenRead();
try {
    Stream strm = reqFTP.GetRequestStream();
    contentLen = fs.Read(buff, 0, buffLength);
    while (contentLen != 0 {
        strm.Write(buff, 0, contentLen);
        contentLen = fs.Read(buff, 0, buffLength);
    }
    strm.Close();
    fs.Close();
}
catch(Exception ex) {
    MessageBox.Show(ex.Message, "Upload Error");
}
}
```

Hỗ trợ FTP trong FtpWebRequest

```
private void Download(string filePath, string fileName) {  
    FtpWebRequest reqFTP;  
    try {  
        FileStream outputStream = new FileStream(filePath + "\\" + fileName, FileMode.Create);  
        reqFTP = (FtpWebRequest)FtpWebRequest.Create(new Uri("ftp://" + ftpServerIP + "/" +  
fileName));  
        reqFTP.Method = WebRequestMethods.Ftp.DownloadFile;  
        reqFTP.UseBinary = true;  
        reqFTP.Credentials = new NetworkCredential(ftpUserID, ftpPassword);  
        FtpWebResponse response = (FtpWebResponse)reqFTP.GetResponse();  
        Stream ftpStream = response.GetResponseStream();  
        long cl = response.ContentLength;
```

Hỗ trợ FTP trong FtpWebRequest

```
int bufferSize = 2048;
int readCount;
byte[] buffer = new byte[bufferSize];
readCount = ftpStream.Read(buffer, 0, bufferSize);
while (readCount > 0) {
    outputStream.Write(buffer, 0, readCount);
    readCount = ftpStream.Read(buffer, 0, bufferSize);
}
ftpStream.Close();
outputStream.Close();
response.Close();
}
catch (Exception ex) {
    MessageBox.Show(ex.Message);
}
}
```

Hỗ trợ FTP trong FtpWebRequest

```
public void DeleteFTP(string fileName)
{
    try {
        string uri = "ftp://" + ftpServerIP + "/" + fileName;
        FtpWebRequest reqFTP;
        reqFTP = (FtpWebRequest)FtpWebRequest.Create( new Uri("ftp://" +
ftpServerIP + "/" + fileName));
        reqFTP.Credentials = new NetworkCredential(ftpUserID, ftpPassword);
        reqFTP.KeepAlive = false;
        reqFTP.Method = WebRequestMethods.Ftp.DeleteFile;
        string result = String.Empty;
```

Hỗ trợ FTP trong FtpWebRequest

```
FtpWebResponse response = (FtpWebResponse)reqFTP.GetResponse();
long size = response.ContentLength;
Stream datastream = response.GetResponseStream();
StreamReader sr = new StreamReader(datastream);
result = sr.ReadToEnd();
sr.Close();
datastream.Close();
response.Close();
}
catch (Exception ex) {
    MessageBox.Show(ex.Message, "FTP 2.0 Delete");
}
}
```

Hỗ trợ FTP trong FtpWebRequest

```
private void btnFileSize_Click(object sender, EventArgs e)
{
    long size = GetFileSize(txtUpload.Text.Trim());
    MessageBox.Show(size.ToString() + " bytes");
}
```


Hỗ trợ FTP trong FtpWebRequest

```
private string[] GetFilesDetailList()
{
    string[] downloadFiles;
    try {
        StringBuilder result = new StringBuilder();
        FtpWebRequest ftp;
        ftp = (FtpWebRequest)FtpWebRequest.Create(new Uri("ftp://" +
ftpServerIP + "/"));
        ftp.Credentials = new NetworkCredential(ftpUserID, ftpPassword);
        ftp.Method = WebRequestMethods.Ftp.ListDirectoryDetails;
```

Hỗ trợ FTP trong FtpWebRequest

```
WebResponse response = ftp.GetResponse();
StreamReader reader = new
StreamReader(response.GetResponseStream());
string line = reader.ReadLine();
while (line != null) {
    result.Append(line);
    result.Append("\n");
    line = reader.ReadLine();
}
result.Remove(result.ToString().LastIndexOf("\n"), 1);
```

Hỗ trợ FTP trong FtpWebRequest

```
        reader.Close();  
        response.Close();  
        return result.ToString().Split('\n');  
    } catch (Exception ex) {  
        System.Windows.Forms.MessageBox.Show(ex.Message);  
        downloadFiles = null;  
        return downloadFiles;  
    }  
}
```

Hỗ trợ FTP trong FtpWebRequest

```
public string[] GetFileList()
{
    string[] downloadFiles;
    StringBuilder result = new StringBuilder();
    FtpWebRequest reqFTP;
    try {
        reqFTP = (FtpWebRequest) FtpWebRequest.Create(new Uri("ftp://" +
ftpServerIP + "/"));
        reqFTP.UseBinary = true;
        reqFTP.Credentials = new NetworkCredential( ftpUserID, ftpPassword);
```

Hỗ trợ FTP trong FtpWebRequest

```
reqFTP.Method = WebRequestMethods.Ftp.ListDirectory;  
WebResponse response = reqFTP.GetResponse();  
StreamReader reader = new StreamReader(  
response.GetResponseStream());  
string line = reader.ReadLine();  
while (line != null) {  
    result.Append(line);  
    result.Append("\n");  
    line = reader.ReadLine();  
}
```

Hỗ trợ FTP trong FtpWebRequest

```
        result.Remove(result.ToString().LastIndexOf('\n'), 1);  
        reader.Close();  
        response.Close();  
        return result.ToString().Split('\n');  
    } catch (Exception ex) {  
        System.Windows.Forms.MessageBox.Show(ex.Message);  
        downloadFiles = null;  
        return downloadFiles;  
    }  
}
```

Hỗ trợ FTP trong FtpWebRequest

```
private void Rename(string currentFilename, string newFilename)
{
    FtpWebRequest reqFTP;
    try {
        reqFTP = (FtpWebRequest)FtpWebRequest.Create(new Uri("ftp://" + ftpServerIP + "/" +
currentFilename));
        reqFTP.Method = WebRequestMethods.Ftp.Rename;
        reqFTP.RenameTo = newFilename;
        reqFTP.UseBinary = true;
    }
```

Hỗ trợ FTP trong FtpWebRequest

```
reqFTP.Credentials = new NetworkCredential(ftpUserID, ftpPassword);  
FtpWebResponse response = (FtpWebResponse)reqFTP.GetResponse();  
Stream ftpStream = response.GetResponseStream();  
ftpStream.Close();  
response.Close();  
} catch (Exception ex) {  
    MessageBox.Show(ex.Message);  
}  
}
```


Hỗ trợ FTP trong FtpWebRequest

```
private void MakeDir(string dirName)
{
    FtpWebRequest reqFTP;
    try {
        reqFTP = (FtpWebRequest)FtpWebRequest.Create(new
        Uri("ftp://" + ftpServerIP + "/" + dirName));
        reqFTP.Method = WebRequestMethods.Ftp.MakeDirectory;
        reqFTP.UseBinary = true;
        reqFTP.Credentials = new NetworkCredential(ftpUserID,
        ftpPassword);
    }
```

Hỗ trợ FTP trong FtpWebRequest

```
FtpWebResponse response = (FtpWebResponse)reqFTP.GetResponse();  
Stream ftpStream = response.GetResponseStream();  
ftpStream.Close();  
response.Close();  
} catch (Exception ex) {  
    MessageBox.Show(ex.Message);  
}  
}
```